

A Novel Approach to Developing Applications in the Pervasive Healthcare Environment through the Use of Archetypes

João Luís Cardoso de Moraes¹, Wanderley Lopes de Souza¹, Luís Ferreira Pires²,
Luciana Tricai Cavalini³, and Antônio Francisco do Prado¹

¹ Federal University of São Carlos, Computer Science, São Carlos, Brazil
{joao_moraes, desouza, prado}@dc.ufscar.br

² University of Twente, Software Engineering Group, Enschede, The Netherlands
l.ferreirapires@utwente.nl

³ Fluminense Federal University, Department of Epidemiology, Niterói, Brazil
lutricav@vm.uff.br

Abstract. Pervasive Healthcare focuses on the use of new technologies, tools, and services, to help patients to play a more active role in the treatment of their conditions. Pervasive Healthcare environments demand a huge amount of information exchange, and specific technologies has been proposed to provide interoperability between the systems that comprise such environments. However, the complexity of these technologies makes it difficult to fully adopt them and to migrate Centered Healthcare Environments to Pervasive Healthcare Environments. Therefore, this paper proposes an approach to develop applications in the Pervasive Healthcare environment, through the use of Archetypes. This approach was demonstrated and evaluated in a controlled experiment that we conducted in the cardiology department of a hospital located in the city of Marília (São Paulo, Brazil). An application was developed to evaluate this approach, and the results showed that the approach is suitable for facilitating the development of healthcare systems by offering generic and powerful approach capabilities.

Keywords: Pervasive Healthcare, Ubiquitous Computing, *openEHR*, Archetypes, Domain Specific Language.

1 Introduction

Most countries face many problems related to healthcare, such as expensive care and the low quality of health services. These problems began to appear as a result of population increase and the lack of healthcare professionals, which is worsening due to the increasing number of the elderly, more chronic disease, and the growing demand for new treatments and technologies. For example, in the United States the number of graduated doctors was 15,632 in 1981 and 15,712 in 2001 (increasing by only 0.5%), while in the same period its population rose from 226 million to 281 million (increasing by 24%) [1].

The current healthcare model is centered on employing highly specialized people, located in large hospitals and focusing on acute cases for treatment but needs to change into a distributed model, to produce faster responses and to allow patients to better manage their own health. The centralized healthcare model implies that patients and caregivers have to visit the same facility (a hospital or clinic) for the healthcare services to be delivered, which is often expensive and inefficient. A distributed healthcare model that pervades the daily lives of the citizens is more able to provide less expensive and more effective and timely healthcare, and characterizes Pervasive Healthcare. According to [2], the goal of Pervasive Healthcare is to enable the management of health and well-being by using information and communication technologies to make healthcare available anywhere, at anytime, and to anyone.

Ubiquitous Computing [3] has been considered to represent the new age of Computing and aims to enable the user to easily access and process information from anywhere, at anytime, and using any kind of device. Ubiquitous Computing environments whether in communities, homes, or hospitals can be extremely useful to build a Pervasive Healthcare model. Particularly, for the “soul” of this model to become directed toward the patient, the efficient care of that patient is fundamental. In this manner, it is necessary that the information exchange among the various professionals, doctors, nurses, health experts responsible for patients, is fast, efficient and safe.

The exchange of health information among heterogeneous Electronic Healthcare Record (EHR) systems in pervasive healthcare environments requires communication standards that enable interoperability between such systems. Although Health Level Seven (HL7)¹ is a widely-used international standard for message exchange between heterogeneous Healthcare Information Systems (HIS), it has some well-known limitations for representing clinical knowledge, such as its combined use of structured components and coded terms, which can result in inconsistent interpretations of clinical information [4]. *openEHR*² is a foundation dedicated to research into interoperable EHR, which defined an open architecture based on two-level modeling that separates information from knowledge, therefore addressing some of the limitations of HL7.

This paper proposes an approach to developing applications in the Pervasive Healthcare environment, by using archetype and Domain Specific Languages (DSLs) [5]. DSLs allow the creation of an infrastructure to reuse and generate most of the applications code of the target domain. The remainder of this paper is structured as follows. Section 2 introduces the main concepts about the *openEHR* standard. Section 3 describes the domain specific languages concept. Section 4 presents the proposed approach illustrated with a case study in the Healthcare domain. Section 5 shows the evaluation of the application developed using the proposed approach. Section 6 discusses some related work. Finally, section 7 presents concluding remarks and proposes further work.

¹ <http://www.hl7.org>

² <http://www.openehr.org>

2 openEHR Dual Model

The *openEHR* architecture was developed based on the two-level modeling paradigm, as shown in Fig. 1. At the first level, a common *Reference Model* (RM) was defined in terms of a predefined set of classes that model the structure of an electronic record; and on the second level, specific concepts were defined by restricting the RM classes in terms of so-called *archetypes*, expressed in the Archetype Definition Language (ADL) [6].

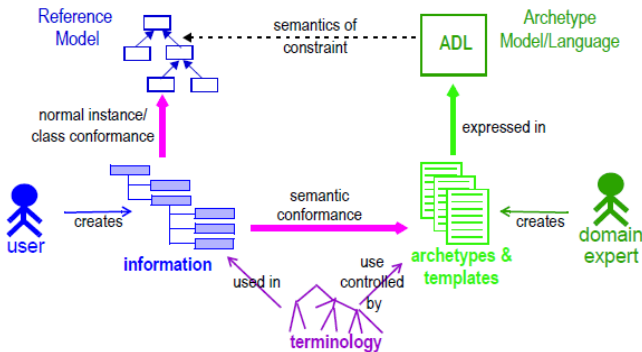


Fig. 1. *openEHR* two-level modeling paradigm

An archetype constitutes a formal model of a domain concept and is expected to be easily understandable by a domain expert. At the implementation level, archetypes can be translated into any language. In accordance with the two-level modeling, data input from users are stored according to the RM, but should also comply with the concepts expressed by the archetypes. The archetypes are designed by the domain experts, and not by information technologies professionals. This split should facilitate the interpretation of the knowledge extracted from the messages exchanged by health systems in various applications.

In the *openEHR* RM, the COMPOSITION class refers to one or more instances of the SECTION class, each containing ENTRY objects. The ENTRY class represents the actual recording of clinical content during a patient Observation, Examination, Assessment, or Intervention. ENTRY is defined as an abstract type with four concrete subtypes: (1) OBSERVATION, which can be used to represent clinical observations, such as blood pressure; (2) EVALUATION, which can be used to represent assessments made after a clinical observation is completed, such as risk assessment; and (3) INSTRUCTION and ACTION, which can be typically used to represent surgical procedures, medication, and other clinical interventions and actions taken. The ACTION subclass describes what was done and committed to the EHR as the result of an INSTRUCTION.

The Clinical Knowledge Manager (CKM) is the archetype repository proposed by the *openEHR* Foundation. This repository contains a set of archetypes that represent clinical concepts and can be reused in various health applications. We have reused

some available archetypes provided by CKM, such as *Device*, *Device Details* and *Clinical Synopsis*, but we have also developed new archetypes to represent clinical concepts of the cardiology domain, such as *Pacemaker Implantation*, *Vascular Cardiac Surgery*, *Coronary Cardiac Surgery*, *Angioplasty Cardiac* and *Pacemaker Evaluation*.

The *openEHR* archetypes were defined for general reuse, and they can accommodate any number of natural languages and terminologies. An archetype consists of three sections: header, definition and ontology. Fig. 2 shows an extract of an ADL archetype for *Pacemaker Implantation* defined according to the *openEHR* standard. The *header* includes the name of the archetype (line 1). The *definition* section contains the structure and restrictions associated with the clinical concept defined by the archetype. *Pacemaker Implantation* specializes the ACTION class (line 4) of the RM. The CLUSTER part (line 8) refers to the stimulations mode for pacemaker implantation, and consists of an ELEMENT with a value of type DV_TYPE. The *ontology* section (line 14-20) includes the terminological definitions. In this example, the linguistic expression ‘Pacemaker Implantation’ is associated with the code ‘at0000’.

```

1  archetype (adl_version=1.4) openEHR-EHR-ACTION.pacemaker_implantation.v1
2  [...]
3  definition
4    ACTION[at0000] matches {--Pacemaker Implantation
5      description matches {
6        ITEM_TREE[at0001] matches { -- Tree
7          items cardinality matches {2..*; unordered} matches {
8            CLUSTER[at0008] occurrences matches {0..1} matches {--Stimulations Mode
9              items cardinality matches {1..2; unordered} matches {
10                ELEMENT[at0003] occurrences {0..1} matches {--Stimulation Mode
11                  value matches {
12                    DV_TEXT matches {*}}
13                [...]
14              ontology
15                term_definitions = <
16                  ["en"] = <
17                    ["at0000"] = <
18                      text = <"Pacemaker Implantation">
19                      description = <"The Pacemaker implant is indicated for patients[...]">
20                    [...]

```

Fig. 2. Archetype for the Pacemaker Implantation concept

The two-level modeling approach applied in *openEHR* allows designers to develop software systems separately from the domain modeling, by specializing and instantiating RM classes. Domain experts can introduce new concepts by defining new archetypes, so that the software systems do not have to be redesigned and redeployed whenever a new concept is defined. Therefore, the two-level modeling approach tends to make the systems easier to maintain and extend with new applications. Archetypes play the role of semantic gateways to terminologies, classifications and computerized clinical guidelines. Archetypes were introduced into *openEHR* to improve the level of semantic interoperability between various healthcare information systems.

3 Domain Specific Languages

A DSL is a language designed to be useful to a specific set of tasks within a given domain [7]. It can be defined by a meta-model, which represents the domain knowledge of the target problem. Restricted to a specific domain, DSLs are usually small, consisting of just a set of abstractions and notations, which are closed to real terms known by the experts of this domain [8]. Thus, DSLs express solutions in the language and abstraction level of the problem domain, reducing the translation efforts of the concepts of this domain into the solution domain.

The use of DSLs in applications modeling, rather than general-purpose modeling languages (e.g., Unified Modeling Language – UML), allows the creation of more specific and complete models. Resources, such as frameworks, design patterns and components, may be included in models, creating an infrastructure that allows the execution of Model-to-Code (M2C) transformations, so as to generate a greater amount of code from the modeling.

4 Proposed Approach

In our approach, we applied *openEHR* dual-level modeling, as a model-driven methodology that has proved in software the achievement of semantic interoperability. The approach combines archetypes and templates in order to create communication interfaces, using ADL as the domain specific language (DSL) that expresses the healthcare domain knowledge.

4.1 Overview

Using the Structured Analysis and Design Technique (SADT) [9] diagram notation, Fig. 3 shows an overview of the approach, which consists of two stages: Domain Engineering (DE) and Application Engineering (AE). In DE a DSL, represented by a

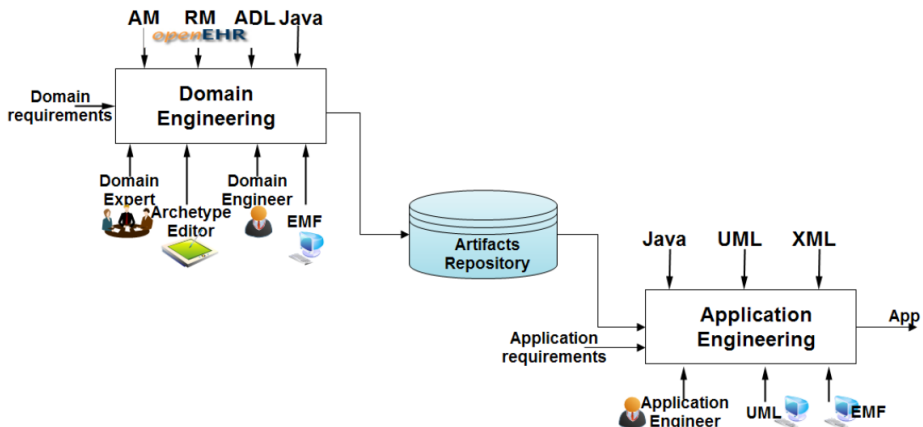


Fig. 3. Overview of the proposed approach

healthcare domain meta-model, and M2C transformations, to code generation are built. In AE, ubiquitous applications are developed through the reuse of the artifacts developed in the DE stage. The meta-model is used to support the application modeling, and the M2C transformations are employed to generate the code that handles *openEHR* messages using archetype.

4.2 Domain Engineering

Fig. 4 illustrates the three activities of the Domain Engineering stage: *Specify Domain Meta-model*, *Design Domain Meta-model*, and *Implement Domain Meta-model*.

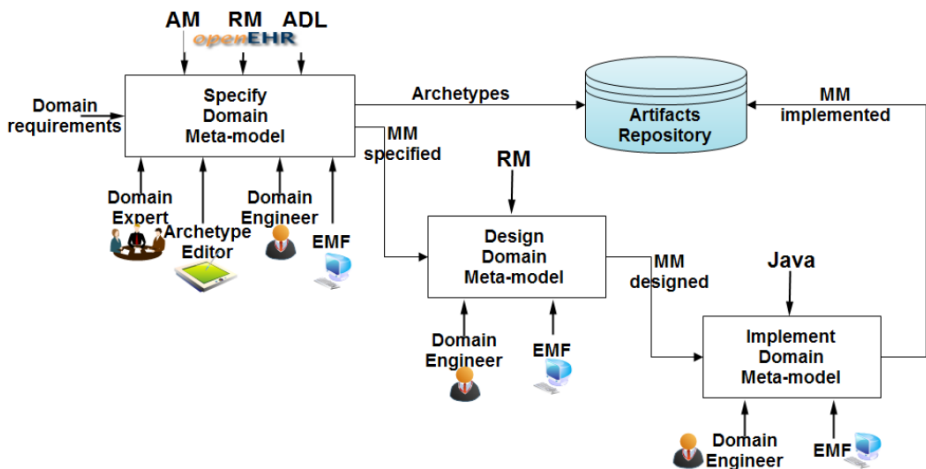


Fig. 4. Overview Domain Engineering

In **Specify Domain Meta-model** activity, the healthcare domain requirements are elicited, specified, analyzed and represented in a meta-model that expresses the knowledge about this domain. In the Eclipse Integrated Development Environment (IDE), the Domain Engineer, guided by the *openEHR* RM specifications, analyses the meta-model. The *openEHR* RM represents the healthcare domain information at a high abstraction level, which allows the use of a range of terminology in the concept description of this domain. The Domain Engineer specifies the archetypes, guided by *openEHR* AM specifications, ADL and the CKM repository. The Domain Experts and the Archetype Editor tool³ are the main mechanisms used to support the archetypes specification. The outputs of domain analysis are domain meta-models, representing the concepts used in domain modeling, and the archetypes specified.

In **Design Domain Meta-model** activity, the meta-model analysis is refined according to standards, technologies and platforms that enable the meta-model's construction. The Domain Engineer, using the Eclipse Modeling Framework (EMF) [10], develops the architecture for the domain meta-model, showing the functional

³ <http://www.oceaninformatics.com/>

decomposition of the domain. The process is guided by medical terminology coding, such as the Systematized Nomenclature of Medicine – Clinical Terms⁴ (SNOMED-CT). SNOMED-CT is a scheme for identifying clinical terms and concepts, and are used by computer systems to support semantic interoperability. Fig. 5 shows part of a designed meta-model with a flexible structure which satisfies all important requirements and still leaves a large degree of freedom for its implementation.

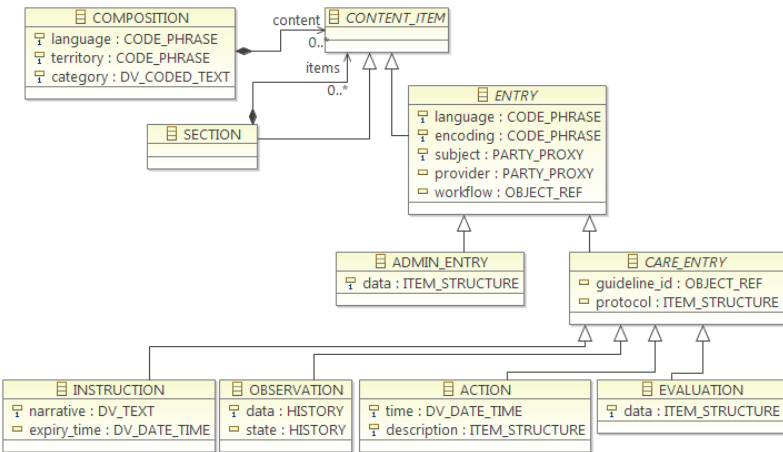


Fig. 5. Portion of openEHR RM meta-model

In **Implement Domain Meta-model** activity, based on the designed meta-model, the implementation is achieved. Supported by the EMF framework, the Domain Engineer automatically generates the Java code of the meta-model and of a model editor. Fig. 6 shows the Java code generation (2) for the ACTION class, based on the designed meta-model (1). This model editor will assist the Domain Engineer during

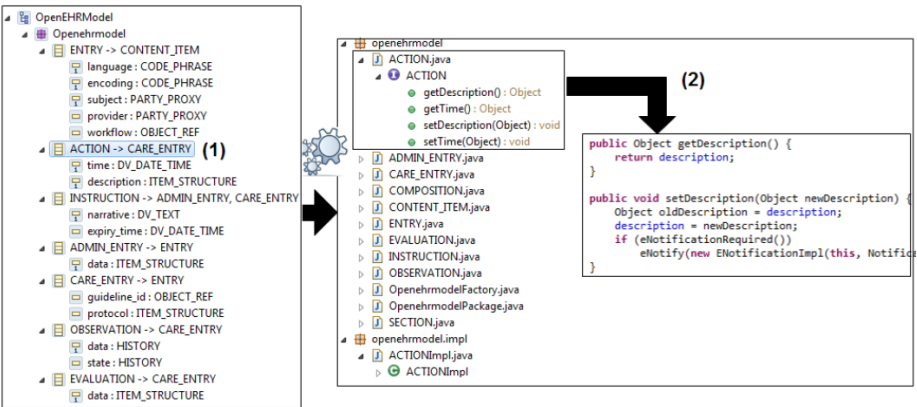


Fig. 6. Sketch of meta-model with Java code generation

⁴ <http://www.ihtsdo.org/snomed-ct/>

the application model specification, during the AE stage. Both the model editor and the meta-model are available as a plugin in Eclipse IDE, ensuring that during AE the meta-model is instantiated to modeling the application. Once the meta-model is based on *openEHR* RM, the models created using DSL expressed by this meta-model, also correspond to refinement that should be performed to create the *openEHR* messages.

4.3 Application Engineering

The AE stage involves the disciplines of Analysis, Design, and Implementation, which are part of the software development process. Fig. 7 shows the three activities of the Application Engineering stage: *Analyzer Application*, *Design Application*, and *Implement Application*.

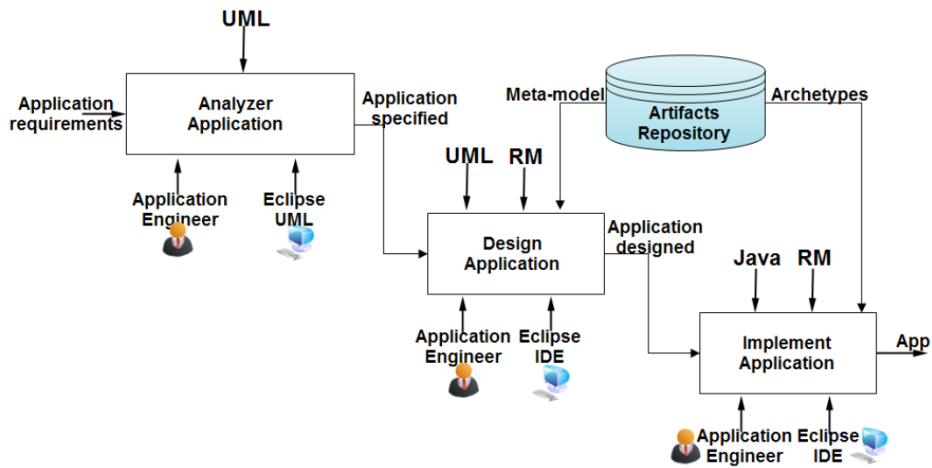


Fig. 7. Overview Application Engineering

The use of DSL expressed by healthcare domain meta-model to application modeling in AE, besides facilitating the application development for this domain, allows mapping the defined process for the *openEHR* messages development into the traditional activities of Software Engineering. Thus, the *openEHR* RM can be performed directly on models that correctly specify the application requirements. Moreover, the code generation automates many of the Application Engineer tasks associated with *openEHR* messages organization, complying with the concepts expressed by the archetypes. The healthcare meta-model also allows the reuse of domain knowledge in various projects during the AE.

To investigate the feasibility of the proposed approach, a case study was developed, building an ubiquitous application that allows evaluation of the implanted pacemaker in patients. The patient receives a notification on his mobile device to schedule an appointment. When the patient arrives in the clinic, the physician receives a message on his device containing information about the patient's pacemaker

implantation. The messages received by the physician and patient are based on *openEHR* archetypes related to the pacemaker implantation concept. The AE activities are detailed below, presenting examples based on the application described as the case study.

In the **Analyzer Application** activity, the application is specified from its requirements, which have been elicited, specified, analyzed and verified. For the specification, the Application Engineer uses UML techniques such as class, sequence and use cases diagrams. An example is the use case diagram shown in Fig. 8, which specifies the requirements to evaluate pacemaker implantation during an appointment. The clinical information was sent to the physician using the *openEHR* standard with archetypes.

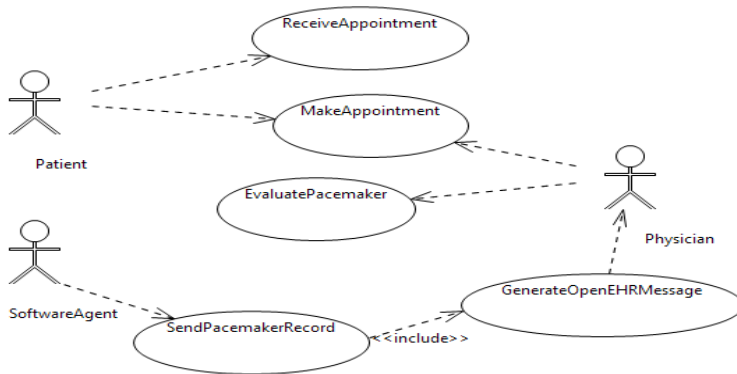


Fig. 8. Application Use Cases

In the **Design Application** activity, the application specification is refined using the technologies of hardware and software platforms, which enable the implementation of the application, such as Java Micro Edition (Java ME). Based on the use case diagrams, in this activity, the Application Engineer also performs the modeling of terms and concepts about the target problem domain as an instance of the meta-model built in DE, selecting the classes and attributes that are relevant to the application domain.

Fig. 9 shows a model built using the model editor plugin available in the Eclipse IDE. This model describes the information associated with the domain applications in accordance with the pacemaker implantation archetype. The *PacemakerImplantation* class is a subtype of the *ACTION* class from *openEHR* RM, as shown in Fig. 9 (1). The attribute *stimulationMode* represents the mode of stimulation used during the pacemaker implantation, according to the at0003 node in the *pacemaker_implantation* archetype, as shown in Fig. 9 (2).

Adopting the coding terminology used in the meta-model (SNOMED-CT), the Application Engineer combines the archetypes that define the clinical concepts represented in the models using the corresponding SNOMED-CT code. For example,

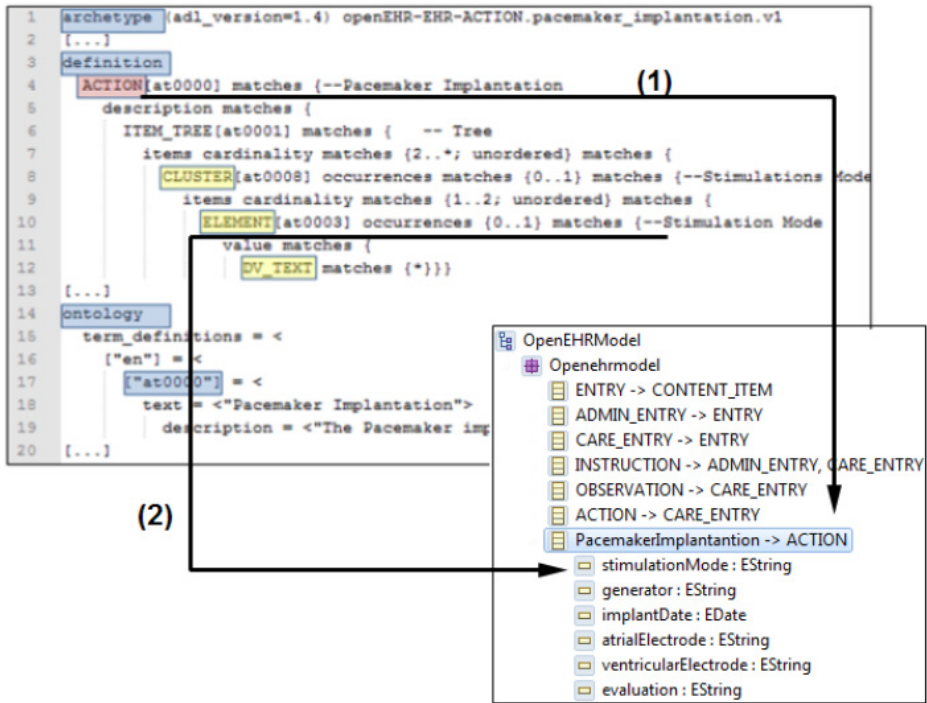


Fig. 9. Meta-model of the Pacemaker Implantation concepts

Fig. 9 shows the *stimulationMode* attribute of the *PacemakerImplantation* class, which was applied the SNOMED-CT code that identifies the pacemaker implantation concept on the archetype ontology section. Thus, the application becomes able to transmit understandable clinical data in an interoperable way.

In the **Implementation Application** activity, the application is implemented, including its communication interface, via integration of archetypes to *openEHR* messages. In the Eclipse IDE, the Application Engineer is assisted by the JET framework to perform the M2C transformations to generate partial code. This code handles the *openEHR* messages using archetypes, which will carry the data related to pacemaker implantation, according to specifications contained in the model. After the partial code generation, the Application Engineer performs the required supplements. Moreover, the other components that comprise the application are developed, such as the user integration interfaces and data persistence. At the end of implementation, tests are performed to provide feedback that indicates whether it is necessary or not to return to the previous activities of AE. Fig. 10 shows the *pacemaker_implantation* archetype and the user interface that contain clinical information related to the case study.

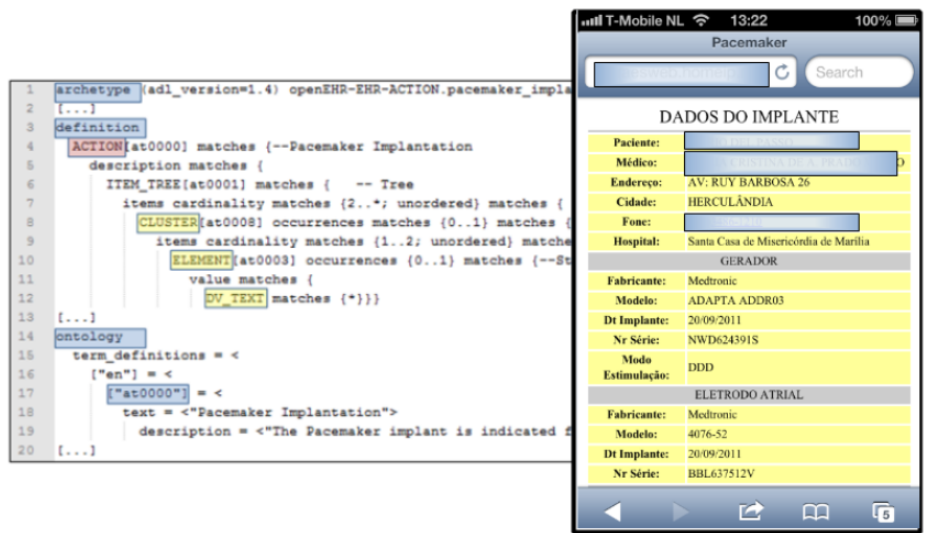


Fig. 10. Pacemaker evaluation – use case

5 Results and Discussion

We conducted a controlled experiment [11] in the cardiology department of the Santa Casa Hospital of Marília (São Paulo, Brazil), using the scenario discussed in Section 4. Table 1 shows the number of participants. The evaluation period was from 1 January 2012 to 30 July 2012.

Table 1. Participants (respondents)

Partici- pants Scenario	Physician	Medical Student	Nurse	Patient	Total
Scenario: Pacemaker Evaluation	3	2	2	15	22

In our study we used the Technology Acceptance Model (TAM) [12], which assumes that *perceived usefulness (PU)* and *perceived ease of use (PEU)* can predict attitudes towards usage and *intentions to use (IU)* technology. TAM is known to be a suitable model for explaining the technology acceptance process in the healthcare sector [13]. To test the effect on technology acceptance, two groups were selected for analysis, involving a limited but still relevant set of people: caregivers (physicians, medical students and nurses) and patients. We decided to group the caregivers together because they work as a team in the scenario.

After the implementation of the scenario, a structured questionnaire based on TAM was sent to the caregivers involved in the scenario, and a questionnaire was also distributed to patients after the medical procedure. Respondents were asked to indicate their agreement or disagreement with several statements on a five-point Likert scale

[14], ranging from 1=strongly disagree to 5=strongly agree. Many empirical studies have demonstrated that the psychometric properties of measurement scales can be affected by the ordering of items within the questionnaire. To avoid this bias, the questionnaire randomly intermixed items across constructs (PU, PEU, IU), and we conducted a group pretest to ensure that the scales were suitable. About 80% and 73% of the questionnaires distributed to caregivers and patients respectively were completed.

Our analysis consisted of two parts: (1) we tested the validity and reliability of the measurement model using Cronbach's Alpha [15]; (2) the data were analyzed using Structural Equation Modeling (SEM) [16] to test both the research model and the hypotheses. SEM is a statistical method that combines factor analysis and path analysis, enables theory construction and analyses the strengths of relationships among variables.

Table 2 shows the results of the validity tests for all scenarios, in which the internal consistency of the constructs were further evaluated for their reliability. The constructs had Cronbach's Alpha values at least close to the limit of 0.700, which is considered very good. Therefore, we concluded that these constructs are reliable for use in our data analysis. Based on this analysis, we concluded that the mean values showed good validity.

Table 2. Statistics and reliability of constructs

Participants	Constructs	Number of Items	Mean	Cronbach's Alpha
Caregivers	Perceived Ease of Use	4	4.05	0.813
	Perceived Usefulness	5	4.50	0.851
	Intention to Use	3	4.39	0.799
Patients	Perceived Ease of Use	4	4.13	0.790
	Perceived Usefulness	5	4.06	0.862
	Intention to Use	3	3.94	0.764

Next, the validated data were analyzed using SEM. The users' intentions to use the communication systems in each scenario can be explained or predicted based on their perception of ease of use and usefulness. Fig. 11 summarizes the research model used in this study. The labels on the arrows show the hypotheses and the path coefficients that show the relative strength and together indicate the causal relationships among variables, whereas R^2 is the percentage of total variance of the independent variables and indicates the predictability of the research model.

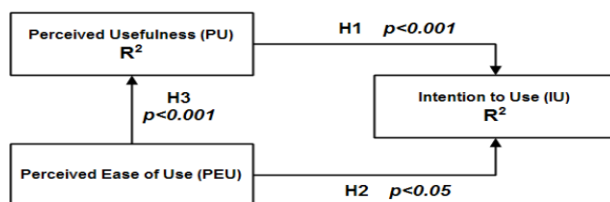


Fig. 11. Research model and hypotheses

The following three hypotheses were proposed and analyzed in each scenario:

H1. Perceived Usefulness positively affects Intention to Use the system.

- Null hypothesis: H_{1n} : $\mu_{\text{PerceivedUsefulness}} = \mu_{\text{IntentionUse}}$
- Alternative hypothesis: H_{1a} : $\mu_{\text{PerceivedUsefulness}} \neq \mu_{\text{IntentionUse}}$

H2. Perceived Ease of Use positively affects Intention to Use the system.

- Null hypothesis: H_{2n} : $\mu_{\text{PerceivedEaseUse}} = \mu_{\text{IntentionUse}}$
- Alternative hypothesis: H_{2a} : $\mu_{\text{PerceivedEaseUse}} \neq \mu_{\text{IntentionUse}}$

H3. Perceived Ease of Use positively affects Perceived Usefulness.

- Null hypothesis: H_{3n} : $\mu_{\text{PerceivedEaseUse}} = \mu_{\text{PerceivedUsefulness}}$
- Alternative hypothesis: H_{3a} : $\mu_{\text{PerceivedEaseUse}} \neq \mu_{\text{PerceivedUsefulness}}$

To verify these three hypotheses, we chose to apply *statistical regression analysis* [17] to the data collected from caregivers and patients in each scenario. The results are summarized in Table 3. For example, in the scenario (participants=caregivers), we test H_{3a} to verify if Perceived Usefulness is determined by Perceived Ease of Use. The details of the regression analysis are: $R^2=0.61$, $p=0.0005$ (very high significance with $p<0.001$ and $\alpha=0.05$). From the result of the regression we could reject the null hypothesis (H_{3n}), meaning that we empirically corroborate that Perceived Usefulness is determined by Perceived Ease of Use, and that H_{3a} was strongly confirmed. R^2 indicates that Perceived Ease of Use explains 61% ($R^2=0.61$) of the variance in Perceived Usefulness. Hypotheses H_{1a} and H_{2a} were confirmed, and Perceived Usefulness strongly determined Intention to Use, and Perceived Ease of Use was a significant secondary determinant.

Table 3. Statistic regression analysis

Scenario	Hypotheses	H1 (PU→IU)		H2 (PEU→IU)		H3 (PEU→PU)	
		R^2	$p<0.001$	R^2	$p<0.05$	R^2	$p<0.001$
Scenario:	Caregivers	0.72	0.0001	0.72	0.009	0.61	0.0005
Pacemaker Evaluation	Patients	0.35	0.0000	0.35	0.007	0.55	0.0002

Therefore, we could conclude that all three hypotheses were confirmed at all points of measurement. Based on this evaluation, the caregivers concluded that the system would be very useful for daily tasks and was very easy to use. Most patients identified some usability benefits, such as the efficient message exchange.

However, our data analysis approach has some limitations. First, the questionnaire model is not completely free of subjectivity for each respondent. Each respondent reacts in a particular way to a questionnaire. Second, we grouped all caregivers together and generalized the results, while we could have split them in several groups. Third, other factors may affect the decision of people to use a given technology, such as their prior experience and job relevance [18], which we did not take into consideration in our work. However, in this study we considered perceived usefulness and perceived ease of use as the most important factors to explain future intention to use the system.

6 Related Work

Due mainly to the complexity involves in the *openEHR* message development process and the lack of comprehensive support tools, various proposals to facilitate the use of the standard can be found in the literature.

In [19], there is a discussion of the implementation of a modeling tool for HL7 messages developed from Eclipse IDE. To handle the artifacts, the developed tool allows serializing of the messages in the XML Metadata Interchange (XMI) standard for later use.

In [20], an approach is presented for developing laboratory information system software by using archetypes to improve the software development process.

In [21], there is a discussion about the implementation of a software factory focused in the healthcare domain. The goal of developing the software factory was to automate the creation of communication interfaces, termed “collaboration ports”, in order to facilitate message exchange.

In [22], a solution is presented to assist developers in handling messages. The authors’ proposal is based on the use of a programmatic DSL to among other features support the messages creation, transmission, interpretation and validation.

The approach proposed in this paper is based on several characteristics of the works described above. It offers its own contributions through the evolution and adaptation of concepts described in related works. Among the contributions of this work, there is the use of archetypes to represent clinical concepts, which are integrated with *openEHR* messages. A DSL is also used that is based on a healthcare domain meta-model, which enables modeling and automatic code generation of the communication interfaces of the applications in that domain.

7 Conclusions

This paper has presented an approach to develop applications in the Pervasive Healthcare environment, and that approach integrates archetypes to *openEHR* messages via DSLs. The integration of the archetypes to *openEHR* messages achieves agility and efficiency in communication between heterogeneous HIS, once the communication becomes interoperable and clinical concepts can be easily identified. DSLs allow the creation of an infrastructure for code generation of these messages, and their reuse in different application of target domain, which reduces the necessary application development effort.

We performed a comprehensive study to investigate the usefulness and ease of use of pervasive healthcare technologies within healthcare environments. We presented the scenario that shows the acceptance of our pervasive healthcare approach by both caregivers and patients, who reacted positively with respect to the usefulness of the architecture. In our approach, the partial code for handling *openEHR* using archetypes messages was generated with the application of suitable M2C transformations.

In future work, we will further evaluate the performance of our approach, especially its scalability, which is a crucial non-functional requirement for realistic applica-

tions and for the simultaneous support of multiple scenarios. We will also measure any improvement in patient safety during the medical procedure. We plan future experiments to extend the TAM models to investigate the effect of other variables - such as, user experience, job relevance and output quality - on perceived usefulness, perceived ease of use, and intention to use.

Acknowledgment. We thank the National Council of Technological and Scientific Development (CNPq) for sponsoring our research in the context of the INCT-MACC.

References

1. Varshney, U.: *Pervasive Healthcare Computing - EMR/EHR, Wireless and Health Monitoring*. Springer, New York (2009)
2. Hansmann, U., Merk, L., Nicklous, M., Stober, T.: *Pervasive Computing: The Mobile World*. Springer (2003)
3. Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. *Commun. ACM* 36, 75–84 (1993)
4. Browne, E.: openEHR Archetypes for HL7 CDA Documents. *Ocean Informatics* (2008)
5. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. *ACM Comput. Surv.* 37, 316–344 (2005)
6. Beale, T., Heard, S.: *Archetype Definition Language*. openEHR Foundation (2007)
7. Gronback, R.: *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit* (2009)
8. van Deursen, A., Klint, P., Visser, J.: Domain-specific languages: an annotated bibliography. *SIGPLAN Not* 35, 26–36 (2000)
9. Ross, D.T.: Structured Analysis (SA): A Language for Communicating Ideas. *IEEE Transactions on Software Engineering* SE-3, 16–34 (1977)
10. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: *EMF: Eclipse Modeling Framework 2*. Addison-Wesley Professional (2009)
11. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. *MIS Quarterly* 28, 75–105 (2004)
12. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q* 13, 319–340 (1989)
13. Chau, P.Y.K., Hu, P.J.H.: Investigating Healthcare Professionals' Decisions to Accept Telemedicine Technology: An Empirical Test of Competing Theories. *Information & Management* 39, 297–311 (2002)
14. Likert, R.: *A Technique for the Measurement of Attitudes*, New York (1932)
15. Nunnally, J.C.: *Psychometric Theory*. Current Contents/Social & Behavioral Sciences (1979)
16. Henseler, J., Chin, W.W.: A Comparison of Approaches for the Analysis of Interaction Effects Between Latent Variables Using Partial Least Squares Path Modeling. *Structural Equation Modeling-a Multidisciplinary Journal* 17, 82–109 (2010)
17. Goldin, R.F.: Review: Statistical Models-Theory and Practice. *The American Mathematical Monthly* 117, 844–847 (2010)
18. Davis, F.D., Venkatesh, V.: Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management. *IEEE Transactions on Engineering Management* 51, 31–46 (2004)

19. Banfai, B., Ulrich, B., Torok, Z., Natarajan, R., Ireland, T.: Implementing an HL7 version 3 modeling tool from an Ecore model. *Studies in Health Technology and Informatics* 150, 157–161 (2009)
20. Piho, G., Tepandi, J., Parman, M., Perkins, D.: From archetypes-based domain model of clinical laboratory to LIMS software. In: *MIPRO, 2010 Proceedings of the 33rd International Convention*, pp. 1179–1184 (2010)
21. Regio, M., Greenfield, J.: *Designing and Implementing an HL7 Software Factory* (2005)
22. Ohr, C., Václavík, M.: Using HL7 Processing Capabilities of the Open Ehealth Integration Platform in the Implementation of IHE Profiles. In: *ICW Developer Conference*, p. 11 (2010)